

Application of material-mesh algebraic collapsing acceleration technique in method of characteristics-based neutron transport code

Ming Dai^{1,2} · Mao-Song Cheng^{1,2}

Received: 16 March 2021/Revised: 11 June 2021/Accepted: 15 June 2021/Published online: 14 August 2021 © China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2021

Abstract The algebraic collapsing acceleration (ACA) technique maximizes the use of geometric flexibility of the method of characteristics (MOC). The spatial grids for loworder ACA are the same as the high-order transport, which makes the numerical solution of ACA equations costly, especially for large-size problems. To speed-up the MOC transport iterations effectively for general geometry, a coarse-mesh ACA method that involves selectively merging fine-mesh cells with identical materials, called material-mesh ACA (MMACA), is presented. The energy group batching (EGB) strategy in the tracing process is proposed to increase the parallel efficiency for microscopic crosssection problems. Microscopic and macroscopic crosssection benchmark problems are used to validate and analyse the accuracy and efficiency of the MMACA method. The maximum errors in the multiplication factor and pin power distributions are from the VERA-4B-2D case with silver-indium-cadmium (AIC) control rods inserted and are 104 pcm and 1.97%, respectively. Compared with the single-thread ACA solution, the maximum speed-up ratio reached 25 on 12 CPU cores for microscopic cross-section VERA-4-2D problem. For the C5G7-2D and

LRA-2D benchmarks, the MMACA method can reduce the computation time by approximately one half. The present work proposes the MMACA method and demonstrates its ability to effectively accelerate MOC transport iterations.

Keywords Algebraic collapsing acceleration · Materialmesh ACA · Method of characteristics · OpenMP · Arbitrary geometry

1 Introduction

Owing to the advantages of excellent geometric flexibility, suitability for large-scale problems and good parallelism, the method of characteristics (MOC) is one of the mainstream methods employed for neutron transport calculation. However, the convergence of scattering source iterations is slow, and the efficient numerical algorithms to speed-up the convergence and multicore parallel technologies are useful. This study focuses on effectively accelerating MOC source iterations for an arbitrary geometry.

The convergence acceleration algorithm generally introduces an efficient low-order approximate solution of the transport equation to speed-up the scattering source iteration, such as the coarse-mesh finite difference (CMFD) [1] and diffusion synthetic acceleration (DSA) [2]. The CMFD method is widely adopted and developed in pressurized water reactors (PWRs). Many improved algorithms have been proposed to enhance the convergence rate or stability, such as partial current-based CMFD (pCMFD) [3], optimally diffusive CMFD (odCMFD) [4] and linear prolongation CMFD (lpCMFD) [5]. The PerMOC code performed CMFD in adjoint mode to accelerate the adjoint

This work was supported by the Chinese TMSR Strategic Pioneer Science and Technology Project (No. XDA02010000) and the Frontier Science Key Program of the Chinese Academy of Sciences (No. QYZDY-SSW-JSC016).

Mao-Song Cheng chengmaosong@sinap.ac.cn

¹ Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China

² CAS Innovative Academies in TMSR Energy System, Chinese Academy of Sciences, Shanghai 201800, China

MOC kernel in the thermal-up-scattering-like iteration scheme [6]. Recently, an equivalent angular flux nonlinear finite difference (ANFD) equation was established to update the MOC incident angular flux and sources directly [7], based on which a novel acceleration technique is expected to be developed to exceed the CMFD performance. Limited by the finite-difference method, these schemes are only suitable for regular geometries. To expand the geometric adaptability of the CMFD method, unstructured CMFD (uCMFD) [8] was developed for unstructured polygonal meshes and generalized CMFD (gCMFD) [9, 10] for general geometries. Nevertheless, the uCMFD method is unsuitable for arbitrary geometries, and gCMFD depends on the width factor adjusted according to the five empirical conditions.

The algebraic collapsing acceleration (ACA) technique [11–15] is a variant of the DSA method. Its solution grid is the same as that of the fine-mesh grid of the transport equation. The ACA equations with the sparse coefficient matrix were constructed approximately from the characteristic line equations. The ACA has good properties in terms of its convergence speed and stability and is suitable for arbitrary geometries. Spectral radius studies [11, 12] showed that convergence can be guaranteed for any optical length; the larger the optical length, the smaller the spectral radius. Divergence due to the large optical length in CMFD can be avoided, and this is also an advantage of the DSA method over CMFD [16]. Owing to the complex proximity relationship between nodes in the unstructured meshes, the number of non-zero elements in the ACA coefficient matrix is increased, which makes the convergence rate of the solution lower than that of the tridiagonal matrix in CMFD. Using the same grids as the transport calculation increases the number of calculations and memory requirements. The efficiency of solving the ACA equations is sensitive to the fine-mesh size, especially in the case of microscopic cross-section problems. In this case, the solution efficiency decreases [12] or even exceeds the ray tracing time. Therefore, it is necessary to study the coarse-mesh method to reduce the computational time and memory requirements of the low-order ACA.

Larsen and Kelley studied the relationship between coarse-mesh DSA (CMDSA) and CMFD [16]. The CMDSA used volume homogenization for the cross-sections in the coarse-mesh cell, and a uniform distribution was used for the flux prolongation of the coarse-mesh cells to the fine-mesh FSRs. For a single track, the ACA equations are equivalent to the characteristic line equations, but the adoption of any homogenization causes them to be inequivalent. Byambaakhuu proposed discontinuous Galerkin DSA (DG-DSA) with coarse-mesh grids, which uses DG discretized coarse-mesh diffusion equations to accelerate the solution of the S_N transport equation with discontinuous finite-element discretization [17]. DG-DSA does not involve cross-section homogenization, but adjusts the mesh size or polynomial order according to the total cross-section of the material. Because the ACA equations are derived from MOC, and the ACA technique has strong geometric adaptability, this idea of broadening the mesh by material mesh is very suitable for ACA. Santandrea studied the DSA acceleration of eigenvalue problems in the MOC [18]. The computational efficiency of the ACA without coarser grids will further deteriorate because the power iteration for solving the ACA equations will take more time.

Because the ACA method is suitable for general geometry, this paper proposes a coarse-mesh ACA by selectively merging some fine-mesh cells with the same material, called material-mesh ACA (MMACA). In this way, the homogenization operation is avoided. The MMACA uses coarse-mesh grids to solve low-order ACA equations, which reduces the number of ACA grids to improve the efficiency of the solution and decreases the size of its coefficient matrix to meet the memory requirements of parallel computing.

The remainder of this paper is organized as follows. In Sect. 2, the basic solution process of the MOC is introduced. The derivation of the elemental equations of the MMACA method is presented in Sect. 3, and the energy group batching (EGB) strategy in the parallel process of ray tracing is presented in Sect. 4. Then, benchmark validation and acceleration performance analyses are presented in Sects. 5 and 6 concludes the paper.

2 Method of characteristics

The steady-state neutron transport equation is written in the following matrix form:

$$\mathbf{L}\Phi = \mathbf{H}\Phi + \frac{\mathbf{F}\Phi}{\lambda},\tag{1}$$

where **L** is the neutron leakage and collision coefficient matrix, **H** is the scattering source coefficient matrix, **F** is the fission source coefficient matrix, λ is the eigenvalue, and **Φ** is the neutron flux. Equation (1) can be transformed into an eigenvalue problem as follows:

$$\mathbf{A}\Phi = \frac{\mathbf{F}\Phi}{\lambda}, \mathbf{A} = \mathbf{L} - \mathbf{H}.$$
 (2)

The power iteration is used to solve Eq. (2):

$$\mathbf{\Phi}^{(k+1)} = \frac{1}{\lambda^{(k)}} \mathbf{A}^{-1} \mathbf{F} \mathbf{\Phi}^{(k)}.$$
(3)

For the multi-group equations of practical problems, the efficiency of the direct inversion of A is low; thus, the

iterative method is usually used. It is transformed into a fixed-source problem to solve

$$\mathbf{A}\Phi^{(k+1)} = \frac{1}{\lambda^{(k)}} \mathbf{F}\Phi^{(k)} = \mathbf{S}^{(k)}.$$
(4)

Equation (4) can be solved by the conventional split iteration method:

$$\mathbf{A}\Phi^{(k)} = (\mathbf{L} - \mathbf{H})\Phi^{(k)} = \mathbf{S}^{(k)}$$

$$\Phi^{(n+1,k)} = \mathbf{L}^{-1}\mathbf{Q}^{(n,k)} = \mathbf{L}^{-1}\mathbf{H}\Phi^{(n,k)} + \mathbf{L}^{-1}\mathbf{S}^{(k)}.$$
(5)
$$\Phi^{(k+1)} = \Phi^{(\infty,k)}$$

The necessary condition for the convergence of Eq. (5) is that the spectral radius of the iteration matrix $\mathbf{P}_{\text{free}} = \mathbf{L}^{-1}\mathbf{H}$ should be smaller than 1.

The MOC approach uses Eq. (3) and Eq. (5) to solve the neutron transport equation. Unlike other methods, MOC does not explicitly construct a matrix L or inverse it. First, the source term is constructed as

$$\mathbf{Q}^{(n,k)} = \mathbf{H} \Phi^{(n,k)} + \mathbf{S}^{(k)}.$$
(6)

Then, based on the integral form of the neutron transport equation, the outgoing segment-boundary angular flux $\phi_{m+1,T}^{(n+1,k)}$ and segment-average angular flux $\overline{\phi}_{m,T}^{(n+1,k)}$ of the *m*-th segment on the characteristic line *T* in the flat source region (FSR) N_m can be obtained from the incoming segment-boundary angular flux $\phi_{m,T}^{(n+1,k)}$ and source term $q_{N_m}^{(n,k)}$:

$$\phi_{m+1,T}^{(n+1,k)} = \alpha_{m,T}\phi_{m,T}^{(n+1,k)} + \frac{1}{4\pi}\beta_{m,T}q_{N_m}^{(n,k)}
\overline{\phi}_{m,T}^{(n+1,k)} = \frac{1}{l_{m,T}} \left[\beta_{m,T}\phi_{m,T}^{(n+1,k)} + \frac{1}{4\pi}\gamma_{m,T}q_{N_m}^{(n,k)}\right],$$
(7)

where $\alpha_{m,T}$, $\beta_{m,T}$ and $\gamma_{m,T}$ are coefficients that are related to the optical length, and $l_{m,T}$ is the length of the *m*-th segment on track *T* in the FSR N_m .

The angular fluxes of all FSRs can be obtained by tracing all the characteristic lines according to Eq. (7). The incoming boundary currents were obtained using the boundary conditions. There is an iterative convergence problem of the boundary currents for a non-vacuum boundary. The required scalar neutron fluxes are obtained by a weighted sum of all angular fluxes, as follows:

$$\phi_{i}^{(n+1,k)} = \frac{1}{V_{i}} \int_{V_{i}} d^{3}r \int_{4\pi} d^{2}\Omega \overline{\phi}_{m,T}^{(n+1,k)}$$

$$= 4\pi \frac{\sum_{p=1}^{P} w_{p} \sum_{m} \delta_{iN_{m}} l_{m,T} \overline{\phi}_{m,T}^{(n+1,k)}}{\sum_{p=1}^{P} w_{p} \sum_{m} \delta_{iN_{m}} l_{m,T}},$$
(8)

where p is the index of the angle discretization, and w_p is the weight of the angle p.

From the above, it can be seen that ray tracing once is equivalent to left multiplying the source term in Eq. (6) by L^{-1} . For practical problems, the split-iteration convergence of Eq. (5) is slow, and the global energy group rebalancing method and the Livolant method [15] can accelerate the convergence to a certain extent, but the effect is limited. In this paper, the ACA method is studied and a coarse-mesh ACA based on a material mesh is proposed.

3 Material-mesh algebraic collapsing acceleration

The ACA method is a variant of DSA. It was derived based on the MOC method, and a low-order neutron transport description similar to the diffusion equation is obtained by an algebraic collapsing approximation, which applies to arbitrary geometry. Because only adjacent FSRs are coupled, the coefficient matrix has good sparsity. The solution of the coefficient matrix can be accelerated using the tracking merging technique (TMT) [19]. This paper proposes a material-mesh ACA method to improve the efficiency of the ACA method.

3.1 Algebraic collapsing acceleration

ACA is a preconditioned Richardson iteration. Equation (5) is left multiplied by L^{-1} , and we obtain

$$\mathbf{B}\Phi^{(k)} = \mathbf{L}^{-1}\mathbf{S}^{(k)}, \mathbf{B} = \mathbf{I} - \mathbf{L}^{-1}\mathbf{H}.$$
(9)

The Richardson iterative method splits the coefficient matrix **B** into $\mathbf{B} = \mathbf{I} - (\mathbf{I} - \mathbf{B})$, and its iterative scheme is written as

$$\Phi^{(n+1/2,k)} = \mathbf{L}^{-1}\mathbf{S}^{(k)} + (\mathbf{I} - \mathbf{B})\Phi^{(n,k)}$$

= $\mathbf{L}^{-1}\mathbf{H}\Phi^{(n,k)} + \mathbf{L}^{-1}\mathbf{S}^{(k)},$ (10)

Equation (10) is the same as Eq. (5). The solution of this free iteration can be obtained using the MOC method. In synthetic acceleration, the new iterative solution is achieved by the additive correction of the free iterative solution:

$$\mathbf{\Phi}^{(n+1,k)} = \mathbf{\Phi}^{(n+1/2,k)} + \mathbf{I}_{\text{int}} \mathbf{\Psi}^{(n+1/2,k)}, \qquad (11)$$

where \mathbf{I}_{int} represents the interpolation matrix between the transport system of the free iteration and the corrective system. It is expected that a corrective flux $\Psi^{(n+1/2,k)}$ will be introduced to make the actual solution converge:

$$(\mathbf{I} - \mathbf{L}^{-1}\mathbf{H})\mathbf{\Phi}^{(n+1,k)} = \mathbf{L}^{-1}\mathbf{S}^{(k)}.$$
(12)

Combining Eq. (10), Eq. (11) and Eq. (12) give the following equation for the corrective flux:

$$(\mathbf{L}-\mathbf{H})\mathbf{I}_{\text{int}}\Psi^{(n+1/2,k)} = \mathbf{H}[\Phi^{(n+1/2,k)} - \Phi^{(n,k)}].$$
 (13)

Equation (13) is a fixed-source problem, similar to Eq. (5). However, the difference is that the source term is replaced by the scattering source term of the flux residual instead of the fission source term. Equation (13) is as difficult to solve as is the case with Eq. (5). The ACA method is used to construct a simplified system of Eq. (13) by using the algebraic collapsing approximation directly on equations for the even symmetric part of the corrective fluxes. These equations have a strongly sparse coefficient matrix, such as diffusion equations and are derived from the basic equations of the MOC method.

In the ACA method, the corrective angular flux is decomposed into an even symmetric part $\psi_{m,T}^{S}$ and an odd symmetric part $\psi_{m,T}^{A}$ [12] as follows:

$$\psi_{m,T}^{S} = \frac{1}{2} (\psi_{m,T} + \psi_{m,-T}) \\ \psi_{m,T}^{A} = \frac{1}{2} (\psi_{m,T} - \psi_{m,-T}),$$
(14)

where -T denotes the direction opposite to the characteristic line *T*. The angular and volume integrals of the even symmetric part will be equal to the corrective scalar flux. The relationship between the segment-averaged angular flux and the segment-boundary angular flux is obtained by eliminating the source term from Eq. (7) for MOC. This is similar to the -T direction. Then, the relationship between the even- and odd-symmetry part of the segment-averaged angular flux and those of the segment-boundary angular flux can be obtained by combining the definition of Eq. (14) as

$$\overline{\psi}_{m,T}^{S} = \frac{1}{2} \left[\psi_{m+1,T}^{S} + \psi_{m,T}^{S} + \tilde{\alpha}_{m,T} \left(\psi_{m+1,T}^{A} - \psi_{m,T}^{A} \right) \right]
\overline{\psi}_{m,T}^{A} = \frac{1}{2} \left[\psi_{m+1,T}^{A} + \psi_{m,T}^{A} + \tilde{\alpha}_{m,T} \left(\psi_{m+1,T}^{S} - \psi_{m,T}^{S} \right) \right], \quad (15)
\widetilde{\alpha}_{m,T} = \frac{2}{1 - \alpha_{m,T}} - \frac{2}{\tau_{m,T}} - 1$$

where $\tau_{m,T}$ is the optical length of the *m*-th segment of characteristic line *T*.

Equation (13) for the corrective flux can be written in a characteristic form and integrating along the m-th segment yields

$$\psi_{m+1,T} - \psi_{m,T} + \tau_{m,T} \overline{\psi}_{m,T} = \frac{l_{m,T}}{4\pi} \hat{q}_{N_m}, \tag{16}$$

where \hat{q}_{N_m} includes fixed and scattering source terms. A similar expression can be obtained for the -T direction. Then, by adding or subtracting two equations, dividing them by two and combining Eq. (14) gives

$$\psi^{A}_{m+1,T} - \psi^{A}_{m,T} + \tau_{m,T} \overline{\psi}^{S}_{m,T} = \frac{l_{m,T}}{4\pi} \hat{q}_{N_m} \\ \psi^{S}_{m+1,T} - \psi^{S}_{m,T} + \tau_{m,T} \overline{\psi}^{A}_{m,T} = 0$$
(17)

By combining Eq. (15) and Eq. (17), we can obtain equations that contain only the even symmetric part of the corrective flux. As an example, take the *m*-th segment, which is away from the boundary surface. If we let the segment in front be j = m-1 and the segment behind be l = m + 1, we can write

$$-\frac{1-b_{j,T}}{d_{ml,T}}\overline{\psi}_{j,T}^{S} + \left[(1-b_{m,T}) \left(\frac{1}{d_{ml,T}} + \frac{1}{d_{jm,T}} \right) + \frac{\tau_{m,T}}{d_{jm,T}d_{ml,T}} \right] \overline{\psi}_{m,T}^{S} - \frac{1-b_{l,T}}{d_{jm,T}} \overline{\psi}_{l,T}^{S} \\ = \frac{b_{j,T}}{d_{ml,T}} \hat{q}_{N_{j}} + \left[-b_{m,T} \left(\frac{1}{d_{ml,T}} + \frac{1}{d_{jm,T}} \right) + \frac{\tau_{m,T}}{d_{jm,T}d_{ml,T}} \right] \hat{q}_{N_{m}} + \frac{b_{l,T}}{d_{jm}} \hat{q}_{N_{l}} \\ \frac{1}{d_{ml,T}} = \frac{1}{d_{m,T}} + \frac{1}{d_{l,T}}, \frac{1}{d_{m,T}} = \frac{\tau_{m,T}}{2 + \tau_{m,T}\tilde{\alpha}_{m,T}} \\ b_{m,T} = \frac{\tau_{m,T}}{2} \left(\frac{1}{d_{m,T}} - \tilde{\alpha}_{m,T} \right) \right)$$
(18)

Equation (18) implies that adjacent regions are connected by characteristic lines, and its coefficient matrix is sparse. Each term of Eq. (18) is integrated as Eq. (8), and the algebraic collapsing approximation is introduced on both sides of the equation. This approximation means that the integral of the product of two functions is approximately the product of their respective integrals, or the assumption of an isotropic even symmetric part of the corrective flux is adopted. For example, for the first term in Eq. (18), we can write

$$\frac{1}{V_{i}} \int_{V_{i}} d^{3}r \int_{4\pi} d^{2}\Omega\left(-\frac{1-b_{j,T}}{d_{ml,T}}\overline{\psi}_{j,T}^{S}\right)$$

$$= \frac{1}{V_{i}} \int_{V_{i}} d^{3}r \int_{4\pi} d^{2}\Omega\left(-\frac{1-b_{j,T}}{d_{ml,T}}\right) \times \frac{1}{V_{i}} \int_{V_{i}} d^{3}r \int_{4\pi} d^{2}\Omega\overline{\psi}_{j,T}^{S}$$

$$= \left[\frac{1}{V_{i}} \int_{V_{i}} d^{3}r \int_{4\pi} d^{2}\Omega\left(-\frac{1-b_{j,T}}{d_{ml,T}}\right)\right] \times \overline{\psi}_{N_{j}}.$$
(19)

After applying an approximation similar to that in Eq. (19) to all the terms in Eq. (18), the equation for the corrective flux can be obtained approximately. This equation is a simplified form of Eq. (13), which can be expressed in the corresponding matrix form as

$$\mathbf{D}\Psi^{(n+1/2,k)} = \mathbf{E}\mathbf{I}_{\mathbf{proj}}[\mathbf{\Phi}^{(n+1/2,k)} - \mathbf{\Phi}^{(n,k)}], \qquad (20)$$

where $\mathbf{I}_{\mathbf{proj}}$ is the projection matrix from the corrective system to the actual neutron transport system. The source term \hat{q}_{N_m} in Eq. (18) includes the corrected flux scattering source term, in addition to the flux residual on the right side of Eq. (20); thus, **D** is related to the coefficients on both sides of Eq. (18). The biconjugate gradient stabilized method (BICGSTAB) with a left incomplete LU (ILU0) precondition can be used to solve Eq. (20). To reduce the non-zero filling items in ILU0 decomposition, the inversion of the breadth-first search (BFS) algorithm can be used to adjust the order of the elements of the corrective flux vector [20].

By combining Eq. (10), Eq. (11) and Eq. (20), we can write

$$\begin{split} \mathbf{\Phi}^{(n+1,k)} &= (\mathbf{I} + \mathbf{I}_{int} \mathbf{D}^{-1} \mathbf{E} \mathbf{I}_{proj}) \mathbf{\Phi}^{(n+1/2,k)} - \mathbf{I}_{int} \mathbf{D}^{-1} \mathbf{E} \mathbf{I}_{proj} \mathbf{\Phi}^{(n,k)} \\ &= \mathbf{\Phi}^{(n,k)} + \mathbf{M} (\mathbf{L}^{-1} \mathbf{S}^{(k)} - \mathbf{B} \mathbf{\Phi}^{(n,k)}) \\ \mathbf{M} &= \mathbf{I} + \mathbf{I}_{int} \mathbf{D}^{-1} \mathbf{E} \mathbf{I}_{proj}. \end{split}$$
(21)

Equation (21) is the standard left-preconditioned Richardson iteration. The preconditioning matrix \mathbf{M} is close to \mathbf{B}^{-1} , and the iterative matrix \mathbf{P}_{ACA} can be written as

$$\mathbf{P}_{ACA} = \mathbf{I} - \mathbf{MB} = \mathbf{P}_{free} - \mathbf{I}_{int} \mathbf{D}^{-1} \mathbf{E} \mathbf{I}_{proj} (\mathbf{I} - \mathbf{P}_{free}).$$
(22)

3.2 Tracking merging technique

The TMT technique [19] compresses the characteristic lines that pass through the same sets of FSRs into one merged track and the contribution of that merged track to the components of the ACA coefficient matrix is then computed. For the *m*-th segment, the weight of the merged track is the sum of the weights of all the compressed characteristic lines, and the length of the merged track is its weighted average value:

$$w_{m,\text{TMT}} = \sum_{T \in \text{TMT}} w_{m,T}$$

$$l_{m,\text{TMT}} = \frac{\sum_{T \in \text{TMT}} w_{m,T} l_{m,T}}{\frac{\sum_{T \in \text{TMT}} w_{m,T} l_{m,T}}{w_{m,\text{TMT}}}}$$
(23)

where TMT represents the merged track. In practical problems, the density of characteristic lines is large, and there is a specific proportion of the characteristic lines that pass through the same sets of FSRs. TMT reduces the number of tracks when solving the coefficient matrix, so it can improve the efficiency with which the coefficient matrix is solved, and it can also reduce the deviation introduced by algebraic collapsing approximation to a certain extent.

3.3 Material-mesh ACA

The basis of the ACA derivation is given by Eq. (15) and Eq. (17), which are derived from Eq. (7) and Eq. (16), respectively, and Eq. (7) is the basic equation of the MOC method and is derived from the integral form of the neutron transport equation. Taking *s* as the distance from any point on the *m*-th segment to the incoming boundary of this

segment, the angular neutron flux of any point on the *m*-th segment can be written as

$$\phi_{m,T}(s) = \phi_{m,T} e^{-\int_{0}^{s} \Sigma(s') \cdot \mathrm{d}s'} + \int_{0}^{s} \mathrm{d}s^{*}q(s-s^{*})e^{-\int_{0}^{s^{*}} \Sigma(s-s') \cdot \mathrm{d}s'},$$
(24)

where s^* is the distance between the source point and the target point.

The premise of Eq. (7), which is derived from Eq. (24), is that the cross-section and source term of the *m-th* segment remain unchanged. To keep the cross-section unaltered, the most direct method is to select the material mesh; that is, FSRs with the same material can be assigned to the same coarse region. Because the coefficient matrix is computed by ray tracing, all FSRs in the same coarse region can form any geometric shape, including a concave shape or a group of discretely distributed shapes. It is worth noting that it is difficult to construct a homogenization method to make the system equivalent before or after homogenization. For example, the optical lengths before and after volume homogenization were markedly different. To make the source term unchanged, M denotes the coarse-mesh segment index on track T and taking the volume-averaged source term of each FSRs in the same coarse region to replace the source term in Eq. (24), which is equivalent to using the algebraic collapsing approximation, it can be expressed as:

$$\phi_{M,T}(s) = \phi_{M,T} e^{-\Sigma_{N_M} \cdot s} + \overline{q}_{N_M} \int_0^s \mathrm{d}s^* e^{-\Sigma_{N_M} \cdot s^*}.$$
 (25)

We can deduce the basic Eq. (7) for the coarse-mesh segments from Eq. (25). Another basic Eq. (16) for the coarse-mesh segments can be derived in the same way. Then, the similar ACA equations that are used in Eq. (20) can be obtained.

The coefficient matrix and equation of the MMACA can be solved with only a few changes in the ACA.

- Before the coefficient matrix is solved, the track information of the fine-mesh grids is transformed into that of the material mesh. The adjacent segments with identical materials are merged, leading to the modification of the number of elements, region numbering and the length of segments.
- 2. When solving the equation, the scalar neutron fluxes of the last iteration and current free iteration are converted into those averaged by the material mesh; thus, the source term computed in each coarse region is the volume-averaged source term of the corresponding FSRs.
- 3. The calculated coarse-mesh correction fluxes were used to correct the free iteration fluxes of all FSRs in

the same coarse region. The uniform source term is used for each FSR in the same coarse region, and then the coarse-mesh correction fluxes were returned to each FSRs homogeneously.

In terms of the matrix-form equations, the interpolation matrix and projection matrix of Eq. (11), Eq. (20) and Eq. (21) are different from those of the original ACA method. In addition to considering the change in the order of elements to reduce the non-zero filling items in ILU0 decomposition, the corresponding mapping between the fine-mesh grids and the coarser material mesh should also be reflected in those two matrices. Owing to the algebraic collapsing approximation, the coarser the grid, the larger will be the deviation. Therefore, the material mesh should be selected properly.

4 Energy group batching strategy in the parallel process of ray tracing

At present, the primary method of parallel computing is to use message passing parallelism between nodes and to use memory-sharing parallelism within a single node. Its purpose is to reduce the memory consumption of a single node and reduce the communication time between the processors. OpenMP is a widely used shared memory programming model that can be controlled by compiling directives, API functions and environmental variables. OpenMP is easy to use, but because of the opacity of interfaces, it is not easy to achieve good parallel scalability. OpenMP has been widely used to parallel MOCs. So far, the mainstream Intel multicore CPU adopts a three-level cache architecture in which the last level of cache L3 is shared. The competition for shared cache L3 is inevitable in multicore parallel, especially for random access to memory data. During ray tracing, the access to FSRs is unordered, which makes the L3 competition important for problems with numerous energy groups. One way to handle this problem is to reduce the working set size of the execution of the core code. A simple and convenient way to achieve this goal is to process ray tracing in energy group batches. To reduce the cost of the repeated construction or destruction of the parallel sections, the loops of the energy group batching (EGB) and the characteristic lines are mixed by the COLLAPSE directive clause to allocate tasks. The pseudo-code is expressed as follows:

!\$OMP PARALLEL &

!\$OMP PRIVATE(...).
!\$OMP DO REDUCTION(+ :phi) COLLAPSE(2).

DO ibatch = 1,nbatch ! the loop of the energy group batching.

DO iline = 1, nline ! the loop of the characteristic lines.

Reading_tracks() ! to get the information of characteristic line.

Tracing_process() ! Ray tracing process through the polar anger loop and energy group loop.

ENDDO ENDDO !\$OMP END DO. !\$OMP END PARALLEL.

The dimension of the scalar flux phi is the number of grids multiplied by the number of energy groups, and its size can reach the capacity of the L3 cache. The REDUCTION directive clause is used in the pseudo-code above to generate copies of variables that need to be modified for each thread, and there will be no pseudosharing problem. However, if the EGB is not adopted, the working set of each thread will be near L3 capacity, and there will be competition in sharing L3 among multiple threads when FSRs are accessed in a nearly random manner. The use of EGB is equivalent to refining the task size. The work originally completed by one thread is further subdivided into N batches, which are completed by different threads in parallel. At this time, the parallel computing work set will be 1/N of the original, leading to a reduction in the competition of the shared L3. Although the cost of repeated construction or destruction of the parallel section can be effectively avoided by a COLLAPSE clause, the addition of an outer loop increases the number of computations, such as repeatedly reading the track information. Then, there is an optimal selection problem for the number of batches N.

5 Benchmark validation and acceleration performance analysis

ThorLAT is a collision probability and MOC-based lattice and burnup code for the analysis of nuclear reactor fuel assemblies. The proposed MMACA method in the ThorLAT code was validated using the VERA-2A, VERA-2F, VERA-4-2D, C5G7-2D and LRA-2D benchmark problems.

The 2A and 2F cases in VERA problems are both assembly geometries, and the 4-2D case is an array of 3×3 assemblies. For a detailed model geometry and parameters, please refer to the literature [21]. The influence of the coarse-mesh partition on the convergence and the effect of the EGB strategy on the acceleration ratio was analyzed using the 1/8 VERA-2A model. We validated the problem with burnable poisons by VERA-2F with complete geometry and studied the acceleration effect of the MMACA method for a larger-size problem. Finally, VERA-4-2D was used to validate the results and performance of the MMACA method for problems with the control rod. In the

current work, the Draglib format database with the SHEM-361 energy group structure generated from ENDF/B-VII.1 [22] was used. Choosing numerous energy groups reduces the influence of the resonance interference effect because the energy group structure is specifically refined. The resonance calculation adopted the subgroup method based on the physical probability table, considered the resonance interference effect by the Bondarenko iteration and ensured the conservation of the reaction rate before and after the subgroup collapsing and volume homogenization by the SPH method. The MOC was used to solve the subgroup slowing down equation.

The C5G7-2D [23] benchmark is a mini-core problem with four 17×17 pin-cell assemblies and five reflector blocks. The LRA-2D [24] benchmark is a 2-group, quartercore transient BWR problem. These macroscopic crosssection benchmark problems can exclude the influence of cross-sections when validating the MMACA method.

All calculations were carried out on a single server that has 32 GB of memory and which uses 12 Intel Xeon silver 4214 CPU cores with a 2.20-GHz main frequency. Some common calculation parameters are as follows: For VERA problems, the characteristic line spacing was 0.05 cm, half of which was used for macroscopic cross-section problems, and the azimuth number was 64. The polar angle number was four, and the convergence accuracy of the eigenvalue was 1.0×10^{-5} . The convergence accuracy of the scale flux was 1.0×10^{-5} for VERA problems and 5.0×10^{-5} macroscopic cross-section problems. A transport-modified P0 cross-section was used. Fixed-source iteration uses the global energy group rebalancing method and the Livolant method in addition to the ACA or MMACA techniques. The Livolant method was also used in power iteration. The diamond difference scheme [25] was used to integrate the characteristic form of the transport equation, and the TMT method was used in computing the ACA coefficient matrix. For macroscopic cross-section problems, only one inner iteration was set up to solve the fluxes.

5.1 1/8 VERA-2A

5.1.1 MMACA validation and analysis

Figure 1 shows the geometric modeling of VERA-2A in ThorLAT. The fuel pin-cell was subdivided into 88 FSRs. The number of FSRs is 3157, and the number of unknowns is approximately 1.0×10^6 . The water gap between the assemblies was explicitly modeled, and the grid of the ACA was the same as that of the FSRs. The different coarse-mesh divisions of the MMACA are shown in Fig. 2. The material mesh of the pin was selected as the coarsemesh in MMACA3, and the number of coarse-mesh cells



Fig. 1 (Color online) Geometric modeling of VERA-2A in ThorLAT



Fig. 2 (Color online) Different coarse-mesh division of MMACA method

was changed from 88 to 4. MMACA1 and MMACA2 represent meshes that are further refined from MMACA3 and the corresponding number of coarse-mesh cells is 16 and 7, respectively. MMACA4 denotes that the cells with two pins having identical materials are merged into the same coarse-mesh cell, which is the case in which the coarse-mesh cell consists of discretely distributed fine-mesh cells. In this case, the number of coarse-mesh cells changed from 176 to 4.

The calculation results of VERA-2A with different ACA grids are listed in Table 1. This problem was solved using a single thread, and the EGB strategy was not used. From the errors of k_{eff} and pin power distribution, MMACA can reach the convergence result of ACA, which indicates the correctness of MMACA. With the coarser ACA grid, the

Table 1The calculation resultsof VERA-2A with differentACA grids

ACA	MMACA1	MMACA2	MMACA3	MMACA4
3157	633	318	194	127
-10	-10	-10	-10	-10
0.11	0.11	0.11	0.11	0.11
0.42	0.42	0.42	0.42	0.42
27	31	33	37	39
1151	734	693	732	757
179	58	43	38	35
592	111	65	47	40
51.4	15.1	9.4	6.4	5.3
1	1.57	1.66	1.57	1.52
	ACA 3157 -10 0.11 0.42 27 1151 179 592 51.4 1	ACA MMACA1 3157 633 -10 -10 0.11 0.11 0.42 0.42 27 31 1151 734 179 58 592 111 51.4 15.1 1 1.57	ACA MMACA1 MMACA2 3157 633 318 -10 -10 -10 0.11 0.11 0.11 0.42 0.42 0.42 27 31 33 1151 734 693 179 58 43 592 111 65 51.4 15.1 9.4 1 1.57 1.66	ACAMMACA1MMACA2MMACA33157633318194-10-10-10-100.110.110.110.110.420.420.420.42273133371151734693732179584338592111654751.415.19.46.411.571.661.57

number of ray tracing increases, which is caused by the deviation introduced by the algebraic collapsing approximation. The coarser the grid is, the greater the deviation introduced by this approximation, which makes the approximation of the preconditioning matrix to the inverse of the coefficient matrix less accurate and increases the number of iterations. However, the increase in the number of iterations is limited. From the perspective of the calculation efficiency, the four MMACA grids can effectively reduce the calculation time, and the speed-up ratio can reach more than 1.5 when compared with ACA, which shows the effectiveness of MMACA at improving the computational efficiency. There are two aspects of the effect of grid coarsening on computational efficiency. First, coarsening the grid leads to more iterations, which is detrimental to improving computational efficiency; the other is that reducing the number of ACA cells can significantly lower the ACA calculation time. Consequently, MMACA2 with a moderate number of coarse-mesh cells had the highest speed-up ratio.

In the original ACA scheme, the time taken to solve the ACA equations of the low-order system accounts for 51%, which indicates that solving ACA equations is inefficient for problems with medium or above size. The reasons are as follows: (1) the complex neighborhood relationship between nodes in unstructured grids increases the number of non-zero elements of the coefficient matrix, and its solution convergence will be lower than that of a standard tri-diagonal matrix in CMFD; (2) for medium or above size problems, the use of the ACA equations to solve problems with numerous energy groups is time-consuming because the iterative solution of upward scattering in thermal groups takes a lot of time; (3) the contribution of each track to the ACA coefficient matrix can be accumulated by ray tracing based on algebraic collapsing approximation, and the larger the problem size, the more time-consuming is the solution of the coefficient matrix. MMACA can effectively reduce the percentage of the ACA run-time. For example, the percentage of the ACA run-time of MMACA2, which is the optimal coarse-mesh division, can be reduced to less than 10%, which shows the need to introduce MMACA for applying ACA to large-scale problems.

5.1.2 Performance analysis of energy group batching

The process of ray tracing can be divided into two parts. First, the information about the track is read, and the boundary or segment-averaged angular flux along the track is then calculated. The track information only needs to be read once when the energy groups are not processed in batches. In contrast, the time taken to read the track information is the number of batches, which increases the number of calculations. This is also the reason for which EGB is not used in general, but it may not be the optimal method for problems having different sizes. The calculation in this section adopts the MMACA2 model for VERA-2A.

Table 2 shows the effect of the number of energy groups per batch (nEGB) on the computational performance, where the case with 361 energy groups in a batch corresponds to non-batch processing. With the increase in nEGB, the run-time first decreased and then increased, and the speed-up ratio declined. Compared with the case with non-batch processing, the minimum-time case, which has 10 energy groups per batch, increases the parallel speed-up ratio from 3.7 to 6.9. The results indicate that the EGB strategy can improve the calculation efficiency and significantly enhance the OpenMP parallel speed-up.

To analyse the impact of the cache on program performance, we used the "perf" tool to record the events of "cache-misses" and "cache-references" in ray tracing. "Cash-misses" can reflect misses of cache-L3, while "cash-references" is the sum of L3 hits and misses, which can reflect L2 misses, as shown in Fig. 3. With the increase in nEGB, the number of L2 and L3 misses of reading track information gradually decreases, while the number of misses of calculations in ray tracing increases. When nEGB is small, the number of cache misses of reading is higher **Table 2** The effect of thenumber of energy groups perbatch on the computationalperformance

Number of energy groups per batch	Single thread (s)	12 Threads (s)	Speed-up ratio
1	770	106	7.3
5	634	91	7.0
10	617	90	6.9
50	628	95	6.6
100	670	117	5.7
361 (non-batch)	694	186	3.7



Fig. 3 Cache misses in ray tracing

than that of tracing because of the repeated reading of track information. When nEGB increases, the number of times of reading decreases and the working set size of the tracing calculation is enlarged, causing the ratio of cache misses to be more focused on the tracing calculation. The total number of cache misses is the combined result of the two processes mentioned above. It first decreases with a decrease in the number of cache misses of reading and then rises with the increase in the number of cache misses of tracing, which is consistent with the trend observed in the calculation time with nEGB in Table 2. When nEGB is 1 in the single-thread calculation, the time is longer than the case without batching. This is because all energy groups need to read the track information again, and the number of calculations is much larger than the case without batching. When nEGB is 50, the number of cache misses under a single thread is less than that of the case when nEGB is 10, but its calculation time is longer. The reason is that with the increase in nEGB, the working set size of the tracing calculation will exceed L2 capacity, resulting in a rapid increase in the number of L2 misses by two orders of magnitude. When 12 threads are used in parallel, the scalar flux is treated by the REDUCTION clause, which means that each thread generates a copy of the scalar flux, leading to a significant increase in the size of the working set. During the parallel calculation, threads compete for the shared L3. When nEGB increases to a certain extent, L3 becomes saturated. If nEGB continues to grow, the number



4

Fig. 4 Strong parallel speed-up ratio for the case with MMACA2 and 10 nEGB

of L3 misses will increase sharply, which will seriously affect the speed-up ratio.

When nEGB is 10 and the ACA coarse-mesh division adopts MMACA2, the strong parallel speed-up ratio is as shown in Fig. 4, where 24 threads are realized by using hyper-threading technology. The maximum speed-up was 8.0, on a 12-core Intel Xeon silver 4214 CPU. Considering the MMACA and EBG introduced in this paper, the maximum speed-up is 15.0, compared with the original scheme.

5.2 VERA-2F with complete geometry

It can be seen from the above calculations that with an increase in the number of cells for ACA, the percentage of the ACA run-time gradually increases. When using the same fine-mesh grid with the transport calculation in Table 1, the ACA total time exceeds the ray tracing time in the solution of the transport equation. If the problem size is further expanded, it can be predicted that the proportion of the ACA solution time will also increase. Hence, the calculation of VERA-2F with complete geometry was carried out. The scale of the unknown variables in this problem was close to 1.0×10^7 . The calculation results are listed in Table 3. The nEGB is 10 and 12 threads are used in parallel. VERA-2F is an assembly with 24 Pyrex burnable poison rods. The results show that MMACA can correctly handle the calculation of such an assembly. The percentage of the ACA run-time reaches 76%, which is the most timeconsuming part of the entire calculation. The use of four types of coarse grids for MMACA can also effectively improve calculation efficiency. MMACA2 is also the best coarse-mesh partition scheme, which can reduce the percentage of the ACA run-time to 14.1% and its speed-up ratio attains 3.1 when compared with the fine-mesh scheme. The computational efficiency of MMACA3 based on the material mesh of the pin-cell was equivalent to that of MMACA2.

Table 4 shows the effect of nEGB on the computational efficiency of the complete geometry VERA-2F when using MMACA2 and 12 threads in the calculation. The effect of nEGB was similar to that of the 1/8 VERA-2A calculations. When nEGB is 10, the calculation efficiency is better, and the speed-up ratio is 2.6 when compared with that of non-batch processing. The increase in the ACA total time without batching is mainly due to the increase in the ACA coefficient matrix calculation. The ACA coefficient matrix is obtained by accumulating through ray tracing, in which the EGB strategy can also be implemented. Compared with scalar flux, the ACA coefficient matrix

calculation requires more memory space to store non-zero elements in the non-diagonal position, and the speed-up effect of the ACA coefficient matrix calculation is more obvious by performing EGB in parallel computing. The reason for which the percentage of the ACA run-time is lower when EGB is not used is that ray tracing is more time-consuming, which reduces the proportion of ACA.

5.3 VERA-4-2D

5.3.1 1/8 symmetric model calculation

VERA-4-2D is a 3×3 color set. 4A-2D is the case in which the control rods are not inserted, 4B-2D is inserted with AIC control rods, and 4C-2D is inserted with B4C control rods. These cases can be used to validate the calculation with the burnable poison Pyrex and control rods inserted into the guide tubes. The results of the control rod worth calculated using the coarse-mesh division MMACA3 for 1/8 VERA-4-2D are shown in Table 5, and 12 threads are used for the calculation. The maximum deviation of the control rod worth calculated by MMACA3 is -1.12%, which is in good agreement with the reference solution. The error distributions of pin power are shown in Fig. 5. The maximum error for the case with the control rods withdrawn was 0.96% and for the case with the rods inserted, it was 1.97%. The error distribution of the pin power is slightly larger than that of VERA-2A, which indicates that there is a certain deviation when treating scattering anisotropy by the transport-corrected P0 crosssection in the presence of a strong absorber. Therefore, a more accurate anisotropic scattering treatment is needed.

5.3.2 Performance analysis for VERA-4-2D with complete geometry

The size of the unknown variables to be solved in VERA-4-2D with complete geometry is approximately 8.0×10^7 , and the ACA calculation is time-consuming,

Table 3 Calculation results of
complete geometry VERA-2F
with different ACA grids and
using 12 threads

	ACA	MMACA1	MMACA2	MMACA3	MMACA4
Number of cells for ACA	26,236	5196	2282	1415	877
k _{eff} diff. (pcm)	-17	-17	-17	-17	-18
RMS of pin power error dist. (%)	0.29	0.29	0.29	0.29	0.29
MAX of pin power error dist. (%)	0.50	0.50	0.49	0.49	0.49
Ray tracing times	32	36	40	43	52
Total time (s)	1538	577	491	495	571
Time for ACA coefficient matrix (s)	68	26	20	19	18
ACA total time (s)	1174	166	69	40	35
Pct. of ACA run-time (%)	76.3	28.8	14.1	8.1	6.1
Speed-up ratio	1	2.67	3.1	3.1	2.69

Table 4 Effect of nEGB on the computational performance for VERA-2F when using MMACA2 and 12 threads

nEGB	Total time (s)	Time for ACA coefficient matrix (s)	ACA total time (s)	Pct. of ACA run-time (%)
1	558	40	89	15.9
5	494	22	71	14.4
10	491	20	69	14.1
50	539	32	80	14.8
100	758	72	120	15.8
361(non-batch)	1300	102	151	11.6

Table 5Control rod worthcalculated using MMACA3 for1/8VERA-4-2D

	Ref. $k_{\rm eff}$	Ref. worth	MMACA3 k _{eff}	Dev. (pcm)	MMACA3 worth	Dev. (%)
4A-2D	1.01024		1.01103	79		
4B-2D	0.98345	2697	0.98448	104	2667	-1.12
4C-2D	0.98029	3024	0.98107	77	3021	-0.11

Fig. 5 (Color online) Error distributions of pin power calculated using MMACA3 for 1/8 VERA-4-2D





which can be inferred from the situation of VERA-2F with complete geometry. To analyse the impact of MMACA and EGB, the following five cases are calculated: "ACA-1-a" represents fine-mesh ACA calculation in which nEGB is "a" and 1 thread is used; "MMACA3-a-b" denotes that

coarse-mesh MMACA3 is used in the calculation, in which nEGB is "b", and the number of threads used is "a". The results are listed in Table 6. The ACA grid is the same as the grid of FSRs for transport computing, which makes the ACA coefficient matrix occupy a significant amount of

	ACA- 1–361	ACA- 1–10	MMACA3- 1-10	MMACA3- 12-10	MMACA3- 12-361
Number of threads	1	1	1	12	12
nEGB	361	10	10	10	361
Number of cells for ACA	226,332	226,332	11,307	11,307	11,307
Number of non-zero elements of ACA coefficient matrix	1,133,828	1,133,828	40,881	40,881	40,881
Ray tracing times	62	62	91	91	91
Total time (s)	253,013	207,497	67,432	9938	29,318
Time for ACA coefficient matrix (s)	28,315	14,225	3201	335	1554
ACA total time (s)	172,927	158,629	5713	825	2084
Pct. of ACA run-time (%)	68.3	76.4	8.5	8.3	7.1
Speed-up ratio	1	1.2	3.8	25.5	8.6

Table 6 Results for VERA-4-2D with complete geometry

memory. Consequently, restricted by the limit of 32 GB memory, only the results using a single-thread can be provided for fine-mesh ACA calculations. Coarse-mesh division MMACA3 can effectively decrease the number of non-zero elements of the ACA coefficient matrix and can significantly reduce the memory requirement of the ACA calculation, so multi-threading parallel computing can be carried out with limited memory. For the cases that involve the use of one thread, EGB can slightly increase the calculation efficiency and the introduction of MMACA can reduce the total time, although the ray tracing times are increased to a certain extent. The percentage of ACA runtime can be reduced to 8.5%, and the speed-up ratio is 3.8. In 12-thread parallel computing, the speed-up ratio is 8.6 without EBG, but the speed-up ratio can be increased to 25.5 by setting nEBG to 10, which shows that EGB can significantly enhance the parallel efficiency. In summary, MMACA can effectively improve the computing efficiency and reduce the memory requirements for problems with a certain scale.

5.4 C5G7-2D

Similar to the above VERA problems, C5G7-2D was solved using 12 threads in four ways, as shown in Fig. 6. The grid for the ACA is also the mesh for the FSRs. The number of unknowns is approximately 2.1×10^6 , which is much smaller than the number of VERA-4-2D problems. Table 7 gives the results for C5G7-2D, and the error distribution of the pin power calculated using MMACA2 is illustrated in Fig. 7. The reference results were obtained using OpenMOC [26]. The results demonstrated good agreement between the MMACA and the reference OpenMOC solution. By reducing the percentage of the ACA run-time from 61.6% to 5.4%, MMACA2 can



Fig. 6 (Color online) Different coarse-mesh division of MMACA for C5G7-2D

achieve a speed-up of about 1.9, which is the most efficient solution. Because C5G7-2D has strong heterogeneity, MMACA2 with the water region of a pin further divided can effectively reduce the number of ray tracing processes compared with MMACA3. MMACA1 with the fuel region further divided had little influence on decreasing the ray tracing times, which makes it less effective than MMACA2.

5.5 LRA-2D

LRA-2D is a benchmark for the diffusion solver, by using which the reference eigenvalue for the initial steady-

	ACA	MMACA1	MMACA2	MMACA3
Number of cells for ACA	301,716	10,693	7225	3757
k _{eff} diff. (pcm)	-21	-21	-20	-21
RMS of pin power error dist. (%)	0.13	0.11	0.11	0.11
MAX of pin power error dist. (%)	0.34	0.31	0.31	0.31
Number of non-zero elements of ACA coefficient matrix	1,206,516	42,644	33,338	12,512
Ray tracing times	18	22	23	30
Total time (s)	495	265	258	328
ACA total time (s)	305	16	14	11
Pct. of ACA run-time (%)	61.6	6.0	5.4	3.3
Speed-up ratio	1	1.9	1.9	1.5

Table 7 Results for C5G7-2D with different ACA grids and using 12 threads

Reference $k_{eff} = 1.186523$ from OpenMOC



Fig. 7 (Color online) The error distribution of pin power calculated using MMACA2 for C5G7-2D

state problem was k_{eff} =0.99637. In this study, OpenMOC was used to generate the reference eigenvalue for the MOC solver. Each assembly was divided into 32 × 32 squares with a side length of 0.46875 cm. Full core geometry with vacuum boundary conditions was used instead of the quarter-core. The coarse grids for an assembly in MMACA1, MMACA2 and MMACA3 are 10 × 10, 4 × 4 and 1 × 1, respectively. In this case, 12 threads were used, and the convergence criterion for solving the ACA equations was set to 1.0×10^{-7} . As shown in Table 8, the MMACA eigenvalue agrees well with the reference MOC solver solution. MMACA2 can drastically decrease the total ACA time and achieve a speed-up by a factor of 2.3×.

6 Conclusion

To maximize the geometric adaptability of the MOC, the convergence acceleration algorithm also needs to be applied to any geometry, and the ACA technique is an effective method for meeting this requirement. The loworder equations with the sparse coefficient matrix can be established by using algebraic collapsing approximation and can be used to accelerate the convergence of the fixedsource iteration in the MOC. Although the ACA equations are very sparse, it still need to face the problem of large memory requirements and inefficiency when solving largesize problems. This is because the ACA is a type of finemesh DSA method. This work enables the ACA equation to be solved on a coarser mesh.

In the current work, the basic solution process of the MOC is first introduced. After the derivation of the ACA, a coarse-mesh MMACA method based on the material mesh was proposed. This method can be realized with a slight modification to the original scheme of the ACA. Then, the EGB strategy is presented to achieve a better parallel efficiency for microscopic cross-section problems. The correctness and effectiveness of the MMACA method under distinct coarse-mesh partitions and with different problem sizes were analyzed numerically. The performance of the EGB strategy was also studied during the numerical validation. From the analysis of the VERA-2A problems, the cache misses caused by nearly random access to FSRs during the tracing process is the main reason for the decrease in parallel efficiency, and the EGB strategy can lessen them to achieve better parallel efficiency by decreasing the working set size simply and conveniently. The multiplication factor and pin power distributions agree well with the reference solutions. The maximum values are 104 pcm and 1.97%, respectively,

	ACA	MMACA1	MMACA2	MMACA3
Number of cells for ACA	495,616	48,400	7744	484
k _{eff} diff. (pcm)	14	14	15	22
Number of non-zero elements of ACA coefficient matrix	1,993,626	195,882	31,756	2114
Ray tracing times	53	54	56	220
Total time (s)	258	140	111	383
ACA total time (s)	167	41	12	8
Pct. of ACA run-time (%)	64.7	29.3	10.8	2.1
Speed-up ratio	1	1.8	2.3	0.7

Table 8 Results for LRA-2D with different ACA grids and using 12 threads

Reference $k_{\text{eff}} = 1.00075$ from OpenMOC

which occurred in the VERA-4B-2D case with AIC control rods inserted. For microscopic cross-section VERA-4-2D problem, a speed-up ratio relative to a single-thread ACA solution can reach 25 on 12 CPU cores. The MMACA method can reduce the computation time cost by approximately one half for macroscopic cross-section C5G7-2D and LRA-2D benchmarks. The results show that the MMACA method can effectively improve the computing efficiency and reduce memory requirements for problems on a certain scale.

Because the algebraic collapsing approximation is adopted in the ACA, the larger the mesh size, the greater will be the deviation caused by the approximation, which leads to a deterioration of the convergence performance. This issue requires further improvement. At present, the proposed MMACA method only accelerates fixed-source iteration. Therefore, an additional acceleration method may be introduced to accelerate the convergence of the power iteration.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Ming Dai. The first draft of the manuscript was written by Ming Dai, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

- Y. Oshima, T. Endo, A. Yamamoto et al., Impact of various parameters on convergence performance of CMFD acceleration for MOC in multigroup heterogeneous geometry. Nucl. Sci. Eng. 194, 477–491 (2020). https://doi.org/10.1080/00295639.2020. 1722512
- M.L. Adams, E.W. Larsen, Fast iterative methods for discreteordinates particle transport calculations. Prog. Nucl. Energy 40, 3–159 (2002). https://doi.org/10.1016/S0149-1970(01)00023-3
- N.Z. Cho, G.S. Lee, C.J. Park, Partial current-based CMFD acceleration of the 2D/1D Fusion method for 3D whole-core transport calculations. Trans. Am. Nucl. Soc. 88, 594 (2003)

- A. Zhu, M. Jarrett, Y.L. Xu et al., An optimally diffusive Coarse Mesh Finite Difference method to accelerate neutron transport calculations. Ann. Nucl. Energy 95, 116–124 (2016). https://doi. org/10.1016/j.anucene.2016.05.004
- Q. Shen, Y. Xu, T. Downar, Stability analysis of the CMFD scheme with linear prolongation. Ann. Nucl. Energy 129, 298–307 (2019). https://doi.org/10.1016/j.anucene.2019.02.011
- K.H. Hosseinipour, F. Faghihi, Development of the neutronics transport code based on the MOC method and adjoint-forward Coarse Mesh Finite Difference technique: PerMOC code for lattice and whole core calculations. Ann. Nucl. Energy 133, 84–95 (2019). https://doi.org/10.1016/j.anucene.2019.04.050
- L.-X. Liu, C. Hao, Y.L. Xu, Equivalent low-order angular flux nonlinear finite difference equation of MOC transport calculation. Nucl. Sci. Tech. **31**, 141 (2020). https://doi.org/10.1007/s41365-020-00834-2
- K.S. Kim, M.D. Dehart, Unstructured partial- and net-current based coarse mesh finite difference acceleration applied to the extended step characteristics method in NEWT. Ann. Nucl. Energy 38, 527–534 (2011). https://doi.org/10.1016/j.anucene. 2010.09.011
- X.M. Chai, X.L. Tu, W. Lu et al., The powerful method of characteristics module in advanced neutronics lattice code KYLIN-2. J. Nucl. Engi. Rad. Sci. 3(3), 031004 (2017). https:// doi.org/10.1115/1.4035934
- X.M. Chai, D. Yao, K. Wang et al., Generalized Coarse-mesh Finite Difference Acceleration for Method of Characteristics in Unstructured Meshes. Chin. J. Comput. Phys. 27(4), 541–547 (2010).
- R. L. Tellier, A. Hébert, Spectral analysis of an algebraic collapsing acceleration for the characteristics method. Paper Presented at M&C+SNA (Palais des Papes, Avignon, France 12–15 September 2005)
- R. L. Tellier, A. Hébert, An Improved algebraic collapsing acceleration with general boundary conditions for the characteristics method. Nucl. Sci. Eng. 156, 121–138 (2007). https://doi. org/10.13182/NSE07-A2691
- A. Hébert, Acceleration of step and linear discontinuous schemes for the method of characteristics in DRAGON5. Nucl. Eng. Tech. 49, 1135–1142 (2017). https://doi.org/10.1016/j.net.2017.07.004
- A. Hébert, Multigroup neutron transport and diffusion computations. in Handbook of Nuclear Engineering ed. by C.D. Gabriel (Springer, New York, 2010), p. 853–858. https://doi.org/10.1007/ 978-0-387-98149-9
- A. Hébert, Applied Reactor physics. (Presses internationals Polytechnique, 2009)

- E.W. Larsen, B.W. Kelley, The Relationship between the Coarse-Mesh Finite Difference and the Coarse-Mesh Diffusion Synthetic Acceleration Methods. Nucl. Sci. Eng. 178, 1–15 (2014). https:// doi.org/10.13182/NSE13-47
- T. Byambaakhuu, D. Wang, S.C. Xiao, A coarse-mesh diffusion synthetic acceleration method with local hp adaptation for neutron transport calculations. Nucl. Sci Eng. **192**, 208–217 (2018).https://doi.org/10.1080/00295639.2018.1499338
- S. Santandrea. An integral multi-domain DPN operator as acceleration tool for the method of characteristics in unstructured meshes. Nucl. Sci. Eng. 155, 223–235 (2007).
- G.J. Wu, R. Roy, Acceleration techniques for trajectory-based deterministic 3D transport solvers. Ann. Nucl. Energy 30, 567–583 (2003). https://doi.org/10.1016/S0306-4549(02)00094-4
- 20. Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edn.(SIAM, Philadelphia, 2003). https://doi.org/10.1137/1. 9780898718003
- A.T. Godfrey, VERA Core Physics Benchmark Progression Problem Specifications, Rev. 4., CASL-U-2012–0131–004 (Oak Ridge National Laboratory, 2014), https://www.casl.gov/sites/

default/files/docs/CASL-U-2012-0131-004.pdf. Accessed 6 November 2020

- 22. A. Hébert, A. Santamarina, Refinement of the Santamarina-Hfaiedh energy mesh between 22.5 eV and 11.4 keV. paper presented at the Int. Conf. on the Physics of Reactors (Interlaken, Switzerland 14–19 Sept 2008)
- M.A. Smith, E.E. Lewis, B.C. Na, Benchmark on deterministic 2-D MOX fuel assembly transport calculationswithout spatial homogenization. Prog. Nucl. Energy 45, 107–118 (2004). https:// doi.org/10.1016/j.pnueene.2004.09.003
- 24. B.N. Aviles, Development of a variable time-step transient NEW code: SPANDEX. Trans. Am. Nucl. Soc. 68 (1993)
- R.L. Tellier, A. Hébert, On the integration scheme along a trajectory for the characteristics method. Ann. Nucl. Energy 33, 14–15 (2006). https://doi.org/10.1016/j.anucene.2006.07.010
- W. Boyd, S. Shaner, L. Li et al., The OpenMOC method of characteristics neutral particle transport code. Ann. Nucl. Energy 68, 43–52 (2014). https://doi.org/10.1016/j.anucene.2013.12.012